Algorithme incrémental pour les graphes de voisins relatifs

Nasreddine Fergani, Ameur Soukhal

Algorithme incrémental pour les graphes de voisins relatifs

Nasreddine FERGANI, Ameur SOUKHAL

Equipe Ordonnancement et Conduite ERL CNRS 6305 ; Laboratoire d'informatique
Université de Tours ; Polytech Tours 64 av. Jean portalis, 37200 Tours, France
nasreddine.fergani@etu.univ-tours.fr, ameur.soukhal@univ-tours.fr

Mots-clés : *Gros volume de données, Indexation multidimensionnelle, Recherche Opérationnelle, Graphe de proximité, Algorithmes*

1 Introduction

Les graphes de voisinage et de proximité connaissent un usage conséquent ces dernières années comme alternatives aux méthodes traditionnelles d'indexation ou de classification de données multidimensionnelles (pages web, documents, images. etc.). Par exemple, la recommandation à base de contenu de blogs internet peut être modélisée par ce type de graphes. Cela est justifié par leur capacité à exprimer de façon intuitive les notions de base, comme la *similarité* ou le *voisinage.* Dans ce travail, on s'intéresse aux graphes de voisins relatifs, noté RNG (*relative neighborhood graph*)[1]. Les RNG trouvaient leurs principales applications en imagerie ou en DataMining [1]. Du fait de leurs propriétés visuellement intéressantes, des travaux récents montrent également leurs apports pour visualiser et analyser de gros volumes de données [3].

RNG est noté $G = (V, E)$, où V est l'ensemble de n sommets (points) $v_1, v_2, \ldots, v_n$ dans un espace multidimensionnel, muni d'une mesure de distance $d(\cdot, \cdot)$. Deux points v_i, v_j sont dits voisins relatifs si et seulement si : $d(v_i, v_j) \leq \max_{k=1,\ldots,n,k\neq i,j}\{d(v_i, v_k), d(v_j, v_k)\}$ (1). L'arête $(v_i, v_j) \in E$ ssi v_i, v_j sont deux voisins relatifs. Soit $\mathcal{H}(v_i, v_j)$ l'hypersphère de rayon $d(v_i, v_j)$ et de centre v_i. Soit $\mathcal{I}(v_i, v_j)$ l'intersection des deux hypersphères $\mathcal{H}(v_i, v_j)$, $\mathcal{H}(v_j, v_i)$ appelée *lune* de (v_i, v_j). Autrement dit, si $(v_i, v_j) \in E$ alors la lune de (v_i, v_j) ne contient aucun autre sommet. Deux types de RNG existent : i) statique : graphe non incrémental, ie. V est fixe ; ii) dynamique : graphe incrémental où V est dynamique (apparition/suppression de nouveaux sommets). Pour son intérêt pratique, nous proposons une nouvelle approche permettant de construire le RNG dynamique par l'apparition de nouveaux sommets. Cela permet d'actualiser la base de données au fur et à mesure de l'apparition de nouveaux objets, et devant être indexés par rapport à leur voisinage.

Soit v_{n+1} le nouveau sommet. La problématique consiste donc à construire $G' = (V \cup \{v_{n+1}\}, E')$ efficacement à partir de G. A notre connaissance, la meilleure approche de résolution connue est celle proposée dans [2] où les auteurs proposent une heuristique de type *insertion locale* : par l'introduction d'un paramètre de distance ε, une liste de voisins L est établie ; L'ensemble E' est reconstruit de façon naïve, c'est-à-dire on supprime toutes les arêtes dont les deux extrémités sont dans L, puis de nouvelles arêtes respectant l'inégalité 1 sont introduites. Si $L = V$, il s'agit d'une méthode exacte où l'insertion se fait en $O(n^3)$, et donc la construction du RNG à n points peut être établie en $O(n^4)$. Dans ce travail, nous proposons donc une méthode exacte et approchée de performances largement meilleures.

2 Nouvelle approche de construction des RNG

Nous proposons de construire $G' = (V \cup \{v_{n+1}\}, E')$ par l'apparition de v_{n+1} en deux phases.
i) Procédure *suppression* : Dans G, éliminer les arêtes selon la propriété 1 ;

ii) Procédure *reconstruction* : construire E' par l'introduction des arêtes de type (v_{n+1}, v_i). On reconstruit donc G' en reliant le nouveau sommet v_{n+1} à ses voisins relatifs. Contrairement à l'approche connue qui consiste à itérer sur tout sommet v_i de V puis insérer (v_i, v_{n+1}) respectant l'inégalité 1, nous ne considérons que les sommets v_j pour lesquels $d(v_{n+1}, v_j) < d(v_{n+1}, v_i)$.

En effet, nous pouvons montrer la propriété suivante :

Propriété 1 *Pour toute arête* $(v_i, v_j) \in E$*,* $(v_i, v_j) \notin E'$ *ssi* $\max\{d(v_{n+1}, v_i), d(v_{n+1}, v_j)\} < d(v_i, v_j)$.

Algorithme 1 : Procédure *suppression*

Données : Le RNG $G = (V, E), v_{n+1}$
Résultat : E modifié

début
  pour tous les $(v_i, v_j) \in E$ **faire**
    si $\max\{d(v_{n+1}, v_i), d(v_{n+1}, v_j)\} < d(v_i, v_j)$ **alors**
      $E = E \backslash \{(v_i, v_j)\}$
    fin
  fin
  retourner E
fin

Algorithme 2 : Procédure *reconstruction*

Données : $E' = suppression(G, v_{n+1})$
Résultat : RNG $G' = (V', E')$

début
  $V' = V \cup \{v_{n+1}\}$; $\sigma = V$
  renuméroter les sommets de σ selon l'ordre croissant de $d(v_{n+1}, \sigma_i)$
  pour $i = 1$ *à* n **faire**
    $j = 1$
    tant que $(d(v_{n+1}, \sigma_j) < d(v_{n+1}, \sigma_i)$ *et* $d(\sigma_j, \sigma_i) \geq d(v_{n+1}, \sigma_i))$ **faire**
      $j = j + 1$
    fin
    si $d(v_{n+1}, \sigma_j) \geq d(v_{n+1}, \sigma_i)$ **alors**
      $E' = E' \cup \{(v_{n+1}, \sigma_i)\}$
    fin
  fin
  retourner *G'*
fin

La procédure de *suppression* nécessite un temps de calcul proportionnel à $|E|$, borné par $O(n^2)$. Donc la construction d'un graphe RNG à l'aide de notre approche incrémentale est bornée par $O(n^3)$.

L'analyse faite sur des instances réelles proposées dans Liu et al. [3], montre que le nombre d'arêtes du RNG est borné par $O(n)$. En pratique, la procédure *suppression* nécessite donc un temps de calcul borné par $O(n)$.

Nous avons généré des instances aléatoires de tailles identiques à celles présentées dans [2]. Les résultats expérimentaux préliminaires montrent l'efficacité de notre approche par rapport aux méthodes connues (le temps de calcul nécessaire pour construire RNG est au moins 1/4 du temps de calcul des méthodes connues).

Pour des gros graphes, nous proposons une heuristique de type insertion locale selon les critères définis dans [2]. Nous transformons ainsi notre méthode exacte en une heuristique de complexité $O(|L|^2)$. Les premiers tests expérimentaux permettent d'obtenir des résultats de meilleure qualité pour un meilleur temps de calcul en comparaison avec l'approche proposée dans [2].

Références

[1] J.W. Jaromczyk, G.T. Toussaint. Relative neighborhood graphs and their relatives. Proceedings of the IEEE, 80 (9) (1992), pp. 1502–1516.

[2] Hakim Hacid, Tetsuya Yoshida : Neighborhood graphs for indexing and retrieving multidimensional data. J. Intell. Inf. Syst. 34(1) : 93-111 (2010).

[3] Tianyang Liu, Fatma Bouali, Gilles Venturini : EXOD : A tool for building and exploring a large graph of open datasets. Computers & Graphics 39 : 117-130 (2014).